

# Informatique en Degead 1



## Présentation, modalités, objectifs (1/3)

Responsable : Denis Cornaz

denis.cornaz@dauphine.fr

01 44 05 41 83

P 409 bis

<http://www.lamsade.dauphine.fr/~cornaz/Enseignement/UV21-Degead1/>

Université Paris-Dauphine

# Planning

- Du 24 Septembre au 28 Octobre puis du 5 au 11 Novembre :  
Semaines 1-6 (S1A)  
1h30 d'informatique par semaine en salle info :
  - ▶ Semaines 1-2 = prise en main Excel
  - ▶ Semaines 3-6 = Algorithmique en Maple
  - ▶ 3 amphis avec moi = 3, 10 et 24 Octobre (slides sur mon site)
- Du 29 Octobre au 4 Novembre : Semaine de consolidation
- Du 12 Novembre au 23 Décembre : Semaines 7-12 (S1B)  
3h d'informatique par semaine =  
1h30 en salle info + 1h30 en salle de cours
  - ▶ Algorithmique en Maple
- 4 dernières semaines du semestre =  
2 de vacances + 1 de révision + 1 d'examen.

# Contenu de ma page web

../Enseignement/UV21-Degead1/

- du cours très détaillé :  
algorithmique de base, Excel, Maple
- des énoncés d'exercices :  
de travaux dirigés (TD), de travaux pratiques (TP)
- des énoncés d'examen (Annales)

# Evaluation

- Note de Contrôle continu sur 20 : 2 interrogations en TD =  
1 sur 8 (45 min en semaine 4) + 1 sur 12 (1h30 en semaine 8)
- Note d'examen sur 20 en Amphi (1h30 sans aucun document) =  
5 points sur Excel + 15 points sur Maple
- Note finale =  $0.5CC + 0.5E$   
(ou zéro en cas de 6 absences justifiées ou non)

# Objectifs

- se familiariser avec les bases de la technologie informatique que l'on rencontre le plus souvent pour l'instant lorsqu'on est gestionnaire : en l'occurrence Excel
- appréhender les concepts informatiques, les bases de l'algorithmique :
  - ▶ variable informatique (math. + dynamique)
  - ▶ tests (algèbre de Boole, logique)
  - ▶ boucles
  - ▶ structures de donnée (linéaires)
- savoir faire des “petits programmes” : en l'occurrence en Maple

# Les algorithmes

Exemples :

- multiplier deux entiers

$$x \times y$$

- calculer la racine carrée d'un nombre

$$\sqrt{x}$$

- déterminer la plus courte tournée d'un voyageur de commerce
- colorier une carte en n'utilisant que 4 couleurs en tout sans que deux pays limitrophes n'aient la même couleur

## Les algorithmes : multiplication (1/3)

Algorithme appris à l'école élémentaire :

$$11 \times 13 = (1 \times 13) + 10 \times (1 \times 13) = 13 + 130 = 143$$

$$13 \times 11 = (3 \times 11) + 10 \times (1 \times 11) = 33 + 110 = 143$$

## Les algorithmes : multiplication (2/3)

Un autre algorithme manipulant deux entiers  $x$  et  $y$  :

*Diviser  $x$  par 2 en arrondissant à l'inférieur et multiplier  $y$  par 2 jusqu'à ce que  $x$  vaille 1, puis additionner les valeurs prises par  $y$  lorsque  $x$  avait une valeur impaire.*

On obtient, avec  $x = 11$  et  $y = 13$  :

$$\begin{array}{r|cccc} x & 11 & 5 & 2 & 1 \\ y & 13 & 26 & 52 & 104 \end{array}$$

## Les algorithmes : multiplication (2/3)

Un autre algorithme manipulant deux entiers  $x$  et  $y$  :

*Diviser  $x$  par 2 en arrondissant à l'inférieur et multiplier  $y$  par 2 jusqu'à ce que  $x$  vaille 1, puis additionner les valeurs prises par  $y$  lorsque  $x$  avait une valeur impaire.*

On obtient, avec  $x = 11$  et  $y = 13$  :

$x$		11	5	2	1
$y$		13	26	52	104

$$13 + 26 + 104 = 143 = x \times y.$$

Est-ce un hasard ?

## Les algorithmes : multiplication (3/3)

$$14 \times 27 = 4 \times 27 + 10(1 \times 27) = 108 + 270 = 378$$

On obtient, avec  $x = 14$  et  $y = 27$  :

$$\begin{array}{r|cccc} x & 14 & 7 & 3 & 1 \\ y & 27 & 54 & 108 & 216 \end{array}$$

## Les algorithmes : multiplication (3/3)

$$14 \times 27 = 4 \times 27 + 10(1 \times 27) = 108 + 270 = 378$$

On obtient, avec  $x = 14$  et  $y = 27$  :

$$\begin{array}{r|rrrr} x & 14 & 7 & 3 & 1 \\ y & 27 & 54 & 108 & 216 \end{array}$$

$$54 + 108 + 216 = 378.$$

Comment certifier le bon fonctionnement d'un algorithme ?

## Les algorithmes : racine carrée (1/3)

Un algorithme manipulant un entier  $y$  :

*Soit  $x = 10$ . Modifier 5 fois la valeur de  $x$  en lui attribuant à chaque fois la nouvelle valeur de*

$$\frac{x + \frac{y}{x}}{2}$$

Pour  $y = 4$ , on obtient :

●

$$x = 10 \quad 5.2 \quad 2.98 \quad 2.16 \quad 2.00$$

## Les algorithmes : racine carrée (1/3)

Un algorithme manipulant un entier  $y$  :

*Soit  $x = 10$ . Modifier 5 fois la valeur de  $x$  en lui attribuant à chaque fois la nouvelle valeur de*

$$\frac{x + \frac{y}{x}}{2} = \frac{10 + \frac{4}{10}}{2} = \frac{10.4}{2} = 5.2$$

Pour  $y = 4$ , on obtient :

$$x = \begin{matrix} & \bullet & & & & & \\ 10 & 5.2 & 2.98 & 2.16 & 2.00 & & \end{matrix}$$

## Les algorithmes : racine carrée (1/3)

Un algorithme manipulant un entier  $y$  :

*Soit  $x = 10$ . Modifier 5 fois la valeur de  $x$  en lui attribuant à chaque fois la nouvelle valeur de*

$$\frac{x + \frac{y}{x}}{2} = \frac{5.2 + \frac{4}{5.2}}{2} = 2.98$$

Pour  $y = 4$ , on obtient :

$$x = 10 \quad 5.2 \quad 2.98 \quad 2.16 \quad 2.00$$

## Les algorithmes : racine carrée (1/3)

Un algorithme manipulant un entier  $y$  :

*Soit  $x = 10$ . Modifier 5 fois la valeur de  $x$  en lui attribuant à chaque fois la nouvelle valeur de*

$$\frac{x + \frac{y}{x}}{2}$$

Pour  $y = 4$ , on obtient :

$$x = 10 \quad 5.2 \quad 2.98 \quad 2.16 \quad 2.00$$

Pour  $y = 16$ , on obtient :

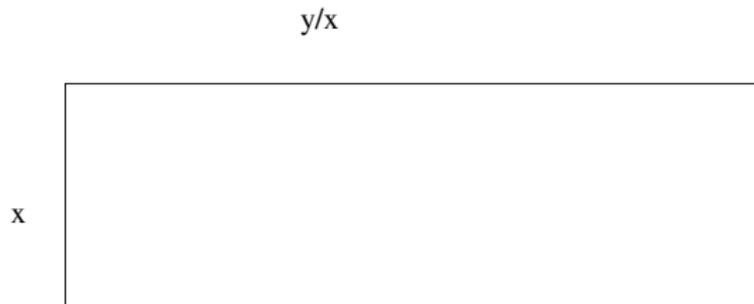
$$x \mid 10 \quad 5.8 \quad 4.28 \quad 4.00 \quad 4.00$$

Pour  $y = 144$ , on obtient :

$$x \mid 10 \quad 12.2 \quad 12.0016 \quad 12.0000 \quad 12.0000$$

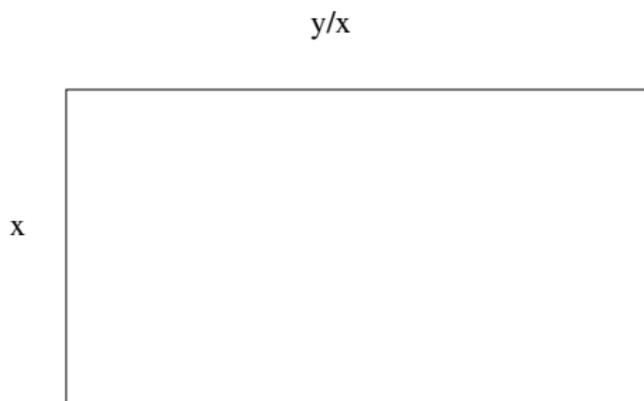
## Les algorithmes : racine carrée (2/3)

Un rectangle dont l'aire vaut  $y$  quelque soit  $x$ , Faire  $x \leftarrow \frac{x + \frac{y}{x}}{2}$



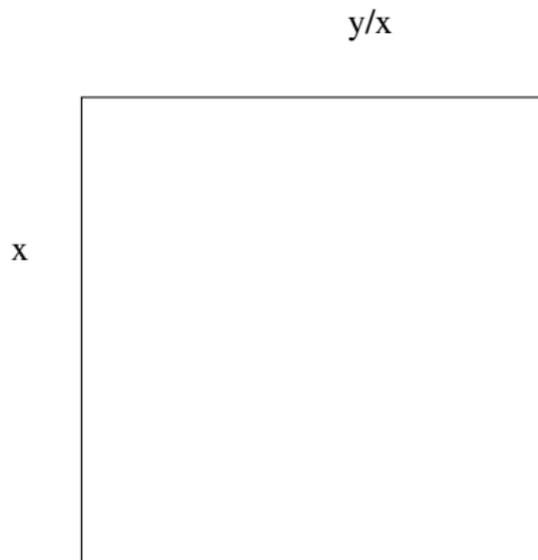
## Les algorithmes : racine carrée (2/3)

Un rectangle dont l'aire vaut  $y$  quelque soit  $x$ , Faire  $x \leftarrow \frac{x + \frac{y}{x}}{2}$



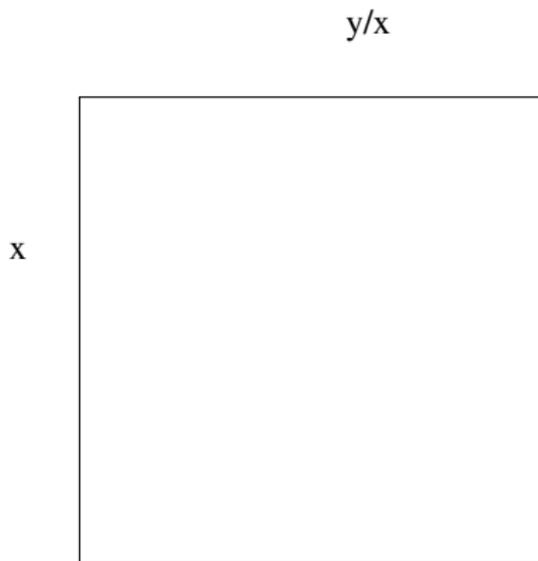
## Les algorithmes : racine carrée (2/3)

Un rectangle dont l'aire vaut  $y$  quelque soit  $x$ , Faire  $x \leftarrow \frac{x + \frac{y}{x}}{2}$



## Les algorithmes : racine carrée (2/3)

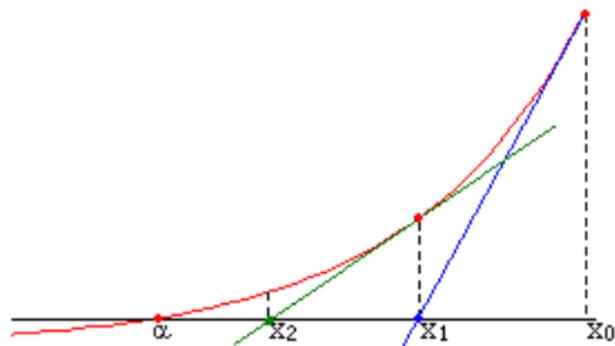
Un rectangle dont l'aire vaut  $y$  quelque soit  $x$ , Faire  $x \leftarrow \frac{x + \frac{y}{x}}{2}$



Si le rectangle est un carré :  $x = \sqrt{y}$

## Les algorithmes : racine carrée (3/3)

$$f(x) = x^2 - y$$

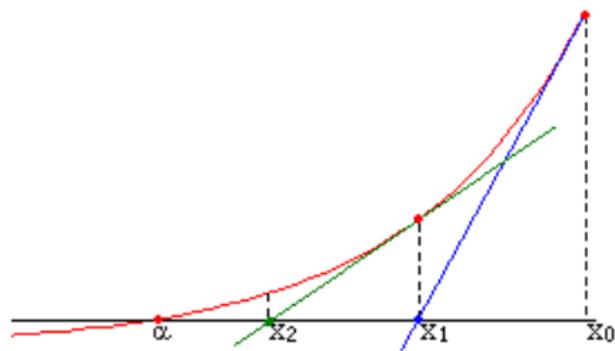


(source : wikipédia)

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \\ &= \lim_{h \rightarrow 0} \frac{(x^2 + 2xh + h^2 - y) - (x^2 - y)}{h} \\ &= \lim_{h \rightarrow 0} 2x + h = 2x \end{aligned}$$

## Les algorithmes : racine carrée (3/3)

$$f(x) = x^2 - y$$



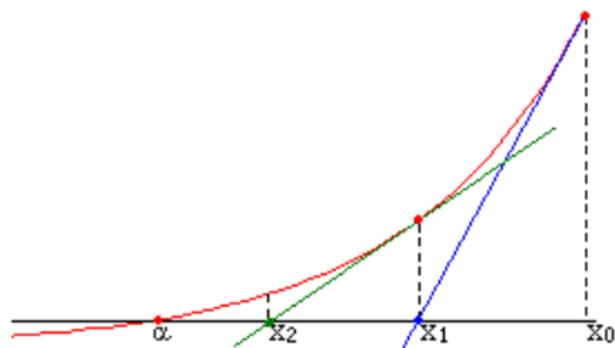
(source : wikipédia)

Suite :  $x_0 = 10$ ,  $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$  :

$$x \leftarrow x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - y}{2x} = \frac{x + \frac{y}{x}}{2}$$

## Les algorithmes : racine carrée (3/3)

$$f(x) = x^2 - y$$



(source : wikipédia)

Suite :  $x_0 = 10$ ,  $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$  :

$$x \leftarrow x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - y}{2x} = \frac{x + \frac{y}{x}}{2}$$

Si  $f(\alpha) = 0$  et  $\alpha \geq 0$  :  $\alpha = \sqrt{y}$ .

## les algorithmes : racine carrée (épilogue)

Aucun algorithme ne donnera la valeur de  $\sqrt{y}$ .

Par exemple si  $y = 2$  :

On ne peut pas stocker dans un ordinateur la valeur de  $\sqrt{2}$  car

$$\sqrt{2} \neq \frac{x}{y} \text{ quelque soient les entiers } x, y.$$

## les algorithmes : racine carrée (épilogue)

Aucun algorithme ne donnera la valeur de  $\sqrt{y}$ .

Par exemple si  $y = 2$  :

On ne peut pas stocker dans un ordinateur la valeur de  $\sqrt{2}$  car

$$\sqrt{2} \neq \frac{x}{y} \text{ quelque soient les entiers } x, y.$$

Sinon il existe  $x, y$  tels que  $x^2 = 2y^2$ .

On peut choisir  $x$  le plus petit possible.

## les algorithmes : racine carrée (épilogue)

Aucun algorithme ne donnera la valeur de  $\sqrt{y}$ .

Par exemple si  $y = 2$  :

On ne peut pas stocker dans un ordinateur la valeur de  $\sqrt{2}$  car

$$\sqrt{2} \neq \frac{x}{y} \text{ quelque soient les entiers } x, y.$$

Mais  $x$  pair si et seulement si  $x^2$  pair,

car  $(2k)^2 = 4k^2$  et  $(2k + 1)^2 = 4k^2 + 4k + 1$ .

- 1  $x$  est pair (donc  $x^2 = 4k^2$ )
- 2  $y$  est pair (car  $y^2 = 2k^2$ )

# Variable informatique

Soit  $x$  une *variable informatique* :

- $x$  a été *déclarée* à l'ordinateur  
(par l'utilisateur ou implicitement selon les langages informatiques)
- $x$  a un *type*, à savoir :  
une place réservée en mémoire + des opérations associées
- on peut modifier la valeur contenue dans  $x$ .

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$x \leftarrow y$$
$$y \leftarrow x$$

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$x \leftarrow y$$

$$y \leftarrow x$$

On obtient, avec  $x = 11$  et  $y = 13$  :

$$\begin{array}{l} x = \\ y = \end{array} \left| \begin{array}{c} 11 \\ 13 \end{array} \right| \begin{array}{l} 13 \\ 11 \end{array}$$

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$z \leftarrow x$$
$$x \leftarrow y$$
$$y \leftarrow z$$

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$z \leftarrow x$$

$$x \leftarrow y$$

$$y \leftarrow z$$

On obtient, avec  $x = 11$  et  $y = 13$  :

$$\begin{array}{l} x = \\ y = \\ z = \end{array} \left| \begin{array}{l} 11 \\ 13 \\ \end{array} \right| \begin{array}{l} 13 \\ \\ 11 \end{array}$$

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$x \leftarrow x + y$$

$$y \leftarrow x - y$$

$$x \leftarrow x - y$$

# Variable informatique

Échange du contenu de deux variables (informatiques)  $x$  et  $y$  :

$$x \leftarrow x + y$$

$$y \leftarrow x - y$$

$$x \leftarrow x - y$$

On obtient, avec  $x = 11$  et  $y = 13$  :

$$\begin{array}{l|l|l} x & 11 & 24 \\ y & 13 & 11 \end{array}$$

# La logique

Les opérateurs fondamentaux : and, or, not

informatique	algèbre de Boole	théorie des ensembles
$x = \text{true}$	$x = 1$	$x \in X$
$x = \text{false}$	$x = 0$	$x \notin X$
$x \text{ and } y$	$x \times y$	$X \cap Y$
$x \text{ or } y$	$x + y$	$X \cup Y$
not $x$	$1 - x$	$\bar{X}$
$x \text{ implies } y$	$x \leq y$	$X \subseteq Y$

Rq. En algèbre de Boole :  $1+1=1$

# La logique

Les opérateurs fondamentaux : and, or, not

informatique	algèbre de Boole	théorie des ensembles
$x = \text{true}$	$x = 1$	$x \in X$
$x = \text{false}$	$x = 0$	$x \notin X$
$x \text{ and } y$	$x \times y$	$X \cap Y$
$x \text{ or } y$	$x + y$	$X \cup Y$
$\text{not } x$	$1 - x$	$\bar{X}$
$x \text{ implies } y$	$x \leq y$	$X \subseteq Y$

Rq. En algèbre de Boole :  $1+1=1$

$x \text{ and not } y$	$x(1 - y)$	$X \setminus Y$
$(x \text{ and not } y) \text{ or } (\text{not } x \text{ and } y)$	$x(1 - y) + (1 - x)y$	$X \Delta Y = (X \setminus Y) \cup (Y \setminus X)$
$(x \text{ or } y) \text{ and not } (x \text{ and } y)$	$(x + y)(1 - xy)$	$X \Delta Y = (X \cup Y) \setminus (X \cap Y)$

## Examen partie Excel 2015

Le tableau Excel ci-dessous doit permettre le calcul du net hors taxe en fonction du brut hors taxe d'une liste clients. Si le brut hors taxe est supérieur à 10000 une remise de 3% ou de 5% est accordée selon le type de client, à savoir 3% aux particuliers et aux détaillants, et 5% aux grossistes.

	A	B	C	D	E	F
1	Nom	Type client	Brut Hors Taxe	Remise	Montant remise	Net Hors Taxe
2	DURANT	Grossiste	15456			
3	DUPONT	Détaillant	9653			
4	MARTIN	Particulier	20000			
5	CARNOT	Grossiste	12456			
6	DAGLAN	Grossiste	9853			
7	MARLOW	Particulier	7562			

	A	B	C	D	E	F
1	Nom	Type client	Brut Hors Taxe	Remise	Montant remise	Net Hors Taxe
2	DURANT	Grossiste	15456			
3	DUPONT	Détaillant	9653			
4	MARTIN	Particulier	20000			
5	CARNOT	Grossiste	12456			
6	DAGLAN	Grossiste	9853			
7	MARLOW	Particulier	7562			

1) Donnez les formules contenues dans les cellules D2, E2, F2 pour qu'on puisse les étendre aux lignes 3 à 7. Par-exemple, MARTIN aura une remise de 3%, soit un montant de 600, et un Net H.T. de 19400.

	A	B	C	D	E	F
1	Nom	Type client	Brut Hors Taxe	Remise	Montant remise	Net Hors Taxe
2	DURANT	Grossiste	15456			
3	DUPONT	Détaillant	9653			
4	MARTIN	Particulier	20000			
5	CARNOT	Grossiste	12456			
6	DAGLAN	Grossiste	9853			
7	MARLOW	Particulier	7562			

2) On calcul ensuite le Net Toutes Taxes Comprises, pour cela la taxe est entrée en G2. Donnez la formule contenue dans le cellule H2 pour qu'on puisse l'étendre aux lignes 3 à 7.

	G	H
1	Taxe	Net T.T.C.
2	10%	
3		
4		
5		
6		
7		

Par-exemple, avec une taxe de 10%, le Net T.T.C. de MARTIN sera 21340.

# Examen partie Maple 2015

*Seuls les opérateurs arithmétiques, la virgule, les crochets, l'affectation :=, et les mots clés suivants sont autorisés :*

- `proc`, `return`, `end`, `if`, `and`, `or`, `not`, `then`, `fi`, `while`, `do`, `od`, `for`, `from`, `to`, `nops`, `NULL`.

*On attend les procédures les plus courtes possibles, et*

- *réutilisant au maximum les procédures*

*des questions précédentes. La longueur des procédures sera pénalisée, typiquement*

- *7-8 lignes*

*est un maximum.*

Étant donnés deux entiers  $a$  et  $b$ , tels que  $a \geq b$ , il existe deux uniques entiers  $q$  (le quotient) et  $r$  (le reste) tels que  $a = qb + r$  avec  $0 \leq r < b$ .

1) Écrire une fonction `MyIrem:=proc(a,b)` prenant deux entiers  $a, b$  ( $a \geq b$ ) en paramètre et qui renvoie la valeur du reste. Écrire aussi une fonction `MyIquo:=proc(a,b)` renvoyant la valeur du quotient.

Par-exemple :

```
> MyIrem(17,5);
```

2

```
> MyIquo(17,5);
```

3

2) Écrire une fonction `MyMember:=proc(L,x)` prenant une liste  $L$  d'entiers et un entier  $x$  en paramètre et retournant `true` si  $x$  est un élément de  $L$ , `false` sinon. Par-exemple :

```
> MyMember([1,5,86,4,2],3);
```

```
false
```

```
> MyMember([1,5,86,4,2],4);
```

```
true
```

3) Écrire une fonction `Clean:=proc(L)` prenant une liste  $L$  d'entiers en paramètre et retournant la liste de tous les entiers distincts de  $L$  :  
Par-exemple :

```
> Clean([4,6,4,1,78,4]);
```

```
[4,6,1,78]
```

4) Écrire une fonction `AllEven:=proc(L)` prenant une liste  $L$  d'entiers en paramètre et retournant la liste de tous les entiers pairs distincts de  $L$  :  
Par-exemple :

```
> AllEven([4,6,89,4,1,75,4,0,56]);  
[4,6,0,56]
```

5) Écrire une fonction `FizBuz:=proc(x)` prenant un entier  $x$  en paramètre et retournant `true` si  $x$  est un multiple de 5 ou de 7 exclusivement, et `false` sinon. Par-exemple :

```
> FizBuz(6);  
false  
  
> FizBuz(15);  
true  
  
> FizBuz(35);  
false
```

Écrire différemment la même fonction, que l'on appellera `FizBuz2:=proc(x)`. La différence doit résider dans la façon d'écrire le test sur les variables booléennes.